Domain Adaptation for Upper Body Pose Tracking in Signed TV Broadcasts

James Charles[1]
j.charles@leeds.ac.uk
Tomas Pfister[2]
tp@robots.ox.ac.uk
Derek Magee[1]
d.r.magee@leeds.ac.uk
David Hogg[1]
d.c.hogg@leeds.ac.uk
Andrew Zisserman[2]
az@robots.ox.ac.uk

[1] School of Computing
University of Leeds
Leeds, UK

[2] Department of Engineering Science
University of Oxford
Oxford, UK

Abstract

The objective of this work is to estimate upper body pose for signers in TV broadcasts. Given suitable training data, the pose is estimated using a random forest body joint detector. However, obtaining such training data can be costly.

The novelty of this paper is a method of transfer learning which is able to harness existing training data and use it for new domains. Our contributions are: (i) a method for adapting existing training data to generate new training data by synthesis for signers with different appearances, and (ii) a method for personalising training data. As a case study we show how the appearance of the arms for different clothing, specifically short and long sleeved clothes, can be modelled to obtain person-specific trackers.

We demonstrate that the transfer learning and person specific trackers significantly improve pose estimation performance.

1 Introduction

We tackle the problem of tracking the upper body pose of people in sign language video. Our source material is signed TV broadcasts incorporating a person translating what is spoken into sign language. The main motivation for tracking the pose is to automatically learn to recognise sign language [2, 6, 8], where upper body layout is of great importance. Tracking is non-trivial due to changing background (the signer is overlaid on the broadcast video, as shown in Figure 1), and also because the signer changes between broadcast and so there are variations in shape and clothing. Furthermore, large quantities of video data are required to learn even a modest number of signs [2, 6], implying the tracker has to be reliable over long video sequences and for multiple signers, and require little or no human supervision.

A number of upper body pose estimators have been applied to this type of data including a method by Fergie *et al*. [9], capable of tracking in real-time, but requiring many frames of video to be annotated and was not shown to generalise to new signers. A tracker by Buehler *et al*. [3] had the capability to produce accurate pose estimates for hours of video sequences, but

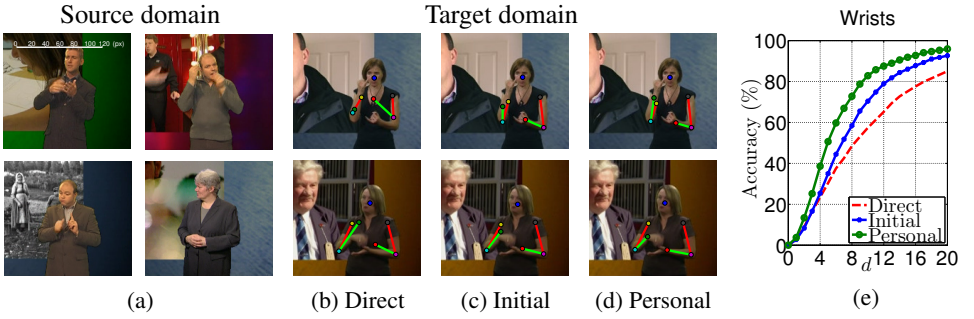


Figure 1: **Training an upper body joint detector using knowledge from the source domain.** In this example, the transfer is from a source domain (a) where signers wear long sleeves to a target domain (b)–(d) where signers wear short sleeves. The red and green lines on the signer show the output pose. The target domain results show: (b) the detector trained directly from the source domain; (c) trained from initial semi-synthetic images constructed using the source domain; and (d) from personal semi-synthetic images constructed from both source and target domain. (e) Accuracy of (b)–(d) in detecting the wrist joint in the target domain. Accuracy is percentage of predicted joints within a distance d pixels from ground truth (scale bar shown in top left of (a)). Note the improvement brought by the two stages – for example the accuracy almost doubles at 8 pixels.

also required manual annotation for initialisation and did not run in real-time. However, due to the extensive amount of tracking output available through this method, Pfister *et al*. [12] were able to use it to train a robust real-time tracking system capable of generalising to new signers provided their appearance was similar to those in training. This was a major limitation though, as it was unable to deal with significant changes in appearance, such as clothing with different sleeve lengths.

In this paper we develop an upper body pose estimator that significantly extends the set of videos that can be used for sign language learning. In general discriminatively trained algorithms do not generalise well to domains that differ from those that they were trained on. We demonstrate how to transfer training data from a source domain to target domain when features are non-transferable. To this end, we make the following contributions: (i) show how to use side-information about a signer's appearance in clothing in the target domain to adapt a tracker, (ii) we create a system for producing semi-synthetic training data of sign language signers with control over both appearance and pose, (iii) an approach for refining the synthesis process to produce personalised upper body trackers, and (iv) we add additional features to the tracker of Pfister *et al*. [12] to improve tracking performance.

In this work we use a random forest body joint detector [12]. This requires large amounts of diverse training data for good generalisation performance [16]. The problem of obtaining sufficient training data was solved in [12] by using a manually trained tracker [3] to annotate training material. However, the tracker was only designed for application on signers wearing long sleeved clothing such as those shown in Figure 1(a). Our target domain will be signers wearing short sleeved clothes, shown in Figure 1(b). We generate training data for the target domain from a source domain of long sleeved clothes in two stages. Stage 1 (Section 2) uses material from the source domain to generate semi-synthetic training data of signers wearing sleeves of a specific length. With this, one can retrain multiple general upper body pose trackers for a particular sleeve length, as shown in Figure 1(c). Stage 2 (Section 3) re-synthesises the training data to become signer specific. Then the refined semi-synthetic data is used to train a personalised tracker tuned to a particular person's arm shape and sleeve length, as shown in Figure 1(d). Each step contributes a significant boost to pose estimation performance, as can be seen in Figure 1(e). The dramatic improvement in joint prediction

accuracy is very visible in a temporal sequence, as shown in the video available online.[1]

Related work. Previous work for sign language recognition in videos has relied on accurate hand tracking. It is popular to use skin colour for hand detection [3, 5, 8, 12, 17], but in most cases this has been applied to videos of signers only wearing long sleeves. Although other detectors based on sliding window classifiers using Haar-like image features [10, 11] have been used, skin colour is a strong cue which should not be ignored. Unfortunately skin colour is a non-transferable feature between signers wearing long sleeves (only showing hand skin) and those wearing short sleeve clothing (showing arm skin). To harness the real-time potential of the upper body detector by Pfister *et al*. [12] for short sleeve signers, we propose to synthesise bare arms onto videos of long sleeved signers where training annotation is available.

Using synthetic images for training pose estimators has been successful in the past [1, 7, 14, 15, 16, 18]. Of particular note is the work by [16] where huge quantities of depth images were synthesised for training a full upper body pose detector. In our case, we adopt a similar approach to generate large amounts of data, but instead combine real data from signers wearing long sleeves and overlay synthetic sleeve information. This method can be thought of as *transfer learning* where knowledge gained from learning one task is transferred to another related task. Interestingly, Farhadi *et al*. [8] used computer-generated avatars to transfer models of sign language to real signers. However, their method of tracking was not learnt from generated data, but rather hand-crafted for signers wearing long sleeves.

2 Synthesising Training Images

Our aim is to train a random forest joint detector, similar to Pfister *et al*. [12], for use in videos where signers are wearing short sleeves. Training of such a system requires thousands of images with annotated body joint locations. Currently the only signed TV broadcast data available in such quantity with annotations (automatically produced [6]) contains signers *only* wearing long sleeves. In this section we show how to transfer knowledge learnt from the data containing long sleeved signers and create thousands of semi-synthetic training images for signers wearing short sleeves. Our system is capable of generating realistic training data with full control over sleeve length as well as pose. Using this data we train a general random forest joint detector tuned for a particular sleeve length.

Input into our joint detector is a colour posterior (CP) image illustrated in Figure 2(a), which is a transformation of the original raw RGB image. The CP image highlights skin, torso and background regions which are important for tracking. To generate semi-synthetic data of signers wearing short sleeves, we combine the realism from CP images of signers wearing long sleeves and augment them with bare arm information. By combining real data with synthetic arm data our synthesiser maintains a greater degree of realism.

Our method for generating semi-synthetic data for tracking a target signer wearing short sleeves takes the following key steps: (i) obtain real colour posterior images and body joint locations of signers wearing long sleeves, (ii) measure the sleeve length of the target signer, (iii) form an arm-skin template for upper and lower arms based upon measured sleeve length, and (iv) synthesise bare arm skin colour on the real CP images according to provided joint locations and arm templates. Details of these steps are given below:

Colour posterior (CP) image. Predicting body joint locations greatly benefits from using information regarding the colour of skin, torso and background. Colour posterior values of the raw RGB images are formed in the same manner as in [12] *i.e*. using skin, torso and

[1] http://www.robots.ox.ac.uk/~vgg/research/sign_language

background colour distributions built from foreground/background segmentation [13] and patches of face and torso detections. A CP image is formed by quantising and scaling the colour posterior values to lie between 0 and 255, with the red, green and blue channel of the image representing the skin, torso and background respectively.

We choose to synthesise CP images over the original RGB image because it contains less information and is therefore easier to model and synthesise accurately. For instance, the CP image removes or at least significantly reduces variations due to texture and lighting effects present in raw RGB image content. Also, the colour distributions used to produce the CP image are fairly peaked, resulting in high confidence for skin, torso and background regions. This has the effect of producing CP images with much smaller variation in pixel values than the original RGB image.

Sleeve length. Sleeve length of a target signer is provided as a value between 0 and 1, which is the normalised length of the sleeve between the signers shoulder and wrist. For example, a sleeve length of 0.5 would represent fully sleeved upper arm and bare lower arm, and a sleeve length of 0.3 would be for a shirt, see Figure 2(a). For a target signer, sleeve length is measured manually on a single image.

Long sleeve data. To produce a large number of synthetic CP images covering a wide range of plausible upper body poses, we obtain thousands of real CP images of long sleeved signers, as shown in Figure 2(b). Upper body joint locations for the head, wrists, elbows and shoulder are provided by an automatic upper body tracker by Buehler *et al*. [3].

Arm template. The appearance of bare arms in a CP image is encapsulated using four rectangular templates describing upper/lower and left/right bare arms in canonical form, as shown in Figure 4. For each pixel in the template, the probability of observing a colour value is stored as an individual colour distribution. Skin regions in the CP image are predominantly red but also contain values from the green and blue channels. These templates can be visualised by displaying the most likely colour value per pixel. In Figure 4 the four initial templates are illustrated in this manner. Notice how bare arm shape is implicitly encoded by the model, with red regions depicting the skin of a bare arm. The initial template is automatically altered to represent a particular sleeve length by simply replacing an upper portion of the template with non-skin distributions.

Arm template initialisation. Initial templates are formed by roughly approximating the shape of fully bare upper and lower arms. The shapes used are crude tapered rectangles from elbow to wrist for lower arms, and shoulder to elbow for upper arms, as shown in Figure 4. All pixels within arm regions represent the same colour distribution. This distribution is learnt from small patches of skin regions in long sleeved signer videos extracted at the head location. Non-skin regions generally contain much less red than green and blue within the CP images. Therefore, we artificially build a colour distribution for non-skin regions in the initial arm template which gives any red colour value greater than 128 (*i.e*. skin colour) a zero probability and equal probability for all other colours.

Synthesising short sleeve data. CP images of signers wearing long sleeves, examples of which are shown in Figure 2(b), are modified so they appear to contain short sleeves of a particular length. Colour values in the CP image are replaced by skin colour posterior values using the arm template which is adjustable for sleeve length. The arm template is positioned according to the provided joint locations and is used as a stencil for placing skin colour values in the correct arm shape, as shown in examples in Figure 2(c). Foreshortening of the arm part is handled naturally by anisotropic scaling of the template in the shoulder/elbow-to-elbow/wrist direction for upper/lower arm parts. Distributions per pixel in the template

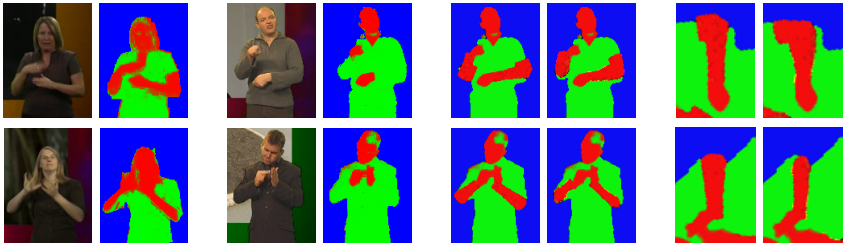


Figure 2: **Stages in synthesising training data.** Rows depict different sleeve length (sleeves in the top row are shorter than in the bottom row). (a) Raw RGB image and CP image counterpart of *short* sleeve signers. (b) Example *long* sleeve signers and CP images. (c) Synthetically produced CP images of short sleeves using CP images from (b) and of initial arm templates. (d) Personalised synthesis using learnt arm templates. For closer comparison, rotated left arms of the synthetic images in (c) and (d) are shown in (e) and (f) respectively.

are used to generate CP colour values. If the colour value is most likely to be skin, *i.e.* has a greater red value than green or blue, the colour is copied to the corresponding pixel of the underlying real CP image. CP colour values are generated by sampling from the distributions independently per pixel in the template. This adds variation to the synthesised CP colour which reduces the risk of the random forest joint detector becoming over-trained on a fixed value.

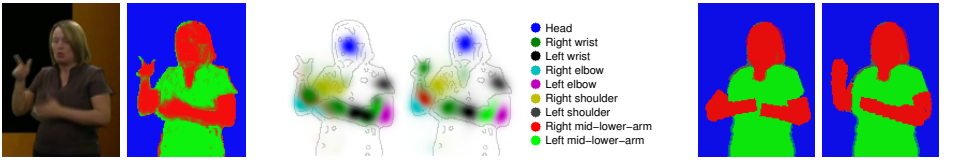
3 Personalising the synthesis

In this section the process of creating semi-synthetic images to automatically train a sleeve length-specific joint detector is taken a stage further. Here, we learn person-specific arm templates to form personalised synthetic training data (depicted in Figure 2) and person-specific upper body trackers. Figure 4 shows examples of personalised arm templates learnt from the signer in Figure 3(a). We show how one can automatically learn arm templates using no manual annotation. This method continually alternates between updating the arm templates and re-training a random forest joint detector. The steps of the method begin as in Section 2 for producing initial semi-synthetic CP images with a sleeve length matching a target signer. The refinement process continues as follows: (i) train a random forest joint detector on semi-synthetic images, (ii) apply the joint detector to a sample of frames of the target signer, (iii) refine the detections using a sample and verify approach, (iv) learn signer-specific arm templates from refined joint locations, and (v) re-synthesise semi-synthetic CP images with personalised arm templates. This process can be repeated from step (i) onwards for further refinement until no change occurs in the arm templates.

3.1 Sample and verify

A random forest body joint detector trained on semi-synthetic images is applied to a set of CP images containing a short sleeved signer, example shown in Figure 3(a)-(b). The random forest provides a confidence map per joint per pixel, see Figure 3(c). Different colours in the map represent different joints, the more intense the colour the higher the likelihood of a joint. The confidence map can be used to infer body joint locations by choosing points which have maximum confidence per joint independently. However the most confident location is not always the correct location and the independence assumption is incorrect.

We address these problems by adopting a sample and verify approach similar to Buehler *et al.* [3] and Charles & Everingham [4]. Joint locations are sampled from the confidence map and scored with a verification function simultaneously using a function which considers the



(a) (b) (c) initial (d) personal (e) (f) initial (g) personal

Figure 3: **Illustration of sample and verify procedure.** (a) Target signer and (b) CP image. (c) Shows body joint confidence map from initial forest, (d) confidence map produced with a personalised forest and (e) the body joint colour key. Most likely whole-image templates using initial (f) and personalised (g) arm templates, computed by sampling joint locations from (c) and (d) respectively. The personalised model produces both better joint samples (most notably for right wrist) and likelihood function.

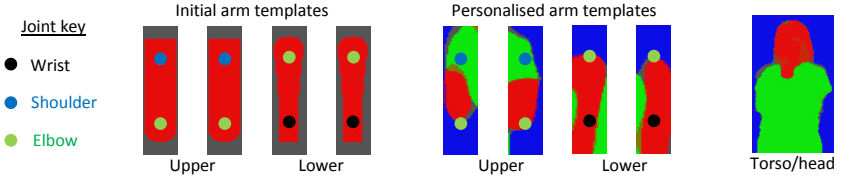


Figure 4: **Initial and personalised arm templates**. The personalised arm and torso/head templates are for the signer in Figure 3(a).

whole image content. This gives a chance to locations with weak confidence plus considers all joints dependently. The sampling process is repeated and the best scoring sample chosen. Although this process is useful for improving joint detections, the refinement process is slow and one loses real-time performance. However, here we only apply the method offline to a sample of frames for more accurately learning arm templates.

3.2 Verification function

Poses sampled from the confidence map are scored using a generative model of the whole CP image. A whole-image template, similar to the arm templates, but of the whole CP image is formed based on a sampled pose. An illustration of a whole-image template is shown in Figure 3(f) for a pose sampled from (c). If x_q is the colour value of a pixel in the CP image at pixel location q, then the probability of observing colour value x_q is given by $p(x_q|\Theta)$, which is the distribution at location q in the whole-image template. Pose is represented by Θ and controls the formation of the whole-image template. The verification scoring function is the log-likelihood of observing the CP image given a pose Θ and is calculated as follows: $L = \sum_{q=1}^{N} \log p(x_q|\Theta)$, where N is total number of pixels in the image. To construct a whole-image template it is first necessary to build a template of the target signer's torso and head.

Torso/head template. A signer-specific torso/head template similar to arm templates is learnt from a training set of CP images of the target signer. Images are aligned according to the head location (provided by a face tracker). A distribution over colour values at each location in the template is built from colour values of corresponding pixels in the training set. A pixel in the training set only contributes to the distribution if lying within a bounding box around the face, or does not represent the arms, *i.e.* more green or blue. An example torso/head template is illustrated in Figure 4, learnt for the signer in Figure 3(a).

Whole-image template. By transforming arm skin and torso templates according to a provided pose Θ one can form a whole-image template, which represents the input CP image. An illustration is shown in Figure 3(f) for a pose sampled from (c). Each pixel in the whole-image template is represented by a colour distribution chosen from one of the five transformed templates. This choice is made per pixel and decided upon in a front to back manner,

based on the depth ordering of the body parts. A depth order is assumed as follows: lower arms are in front of upper arms and upper arms are in front of the torso-head. Distributions in the arm templates are only used if they represent skin *i.e*. the distribution gives highest probability for red colour values than green or blue values.

3.3 Learning arm templates

Arm templates are learnt for a particular signer by detecting joints, using the above approach, on a training set of CP images. Detections are used to extract rectangular windows of upper and lower arm parts and place them in canonical form. This results in four sets of training windows for upper/lower and left/right arm parts. For each arm part, the corresponding training windows are used to learn colour distributions for an arm template. Each pixel in the template represents a distribution which is learnt from colour values of corresponding pixels in the training windows.

Re-training the joint detector. Personalised arm templates are used to retrain a person-specific random forest joint detector. Furthermore, by iterating the above procedure the refinement increases the fidelity of whole-image templates and continually improves the performance of our verification function before reaching saturation in three iterations. Illustration of the improvement in joint confidence maps produced using the initial and personalised random forest are shown in Figure 3(c) and (d) respectively. Figure 3(d) shows tighter confidence close to correct joint locations than in Figure 3(c). The most likely whole-image templates using poses sampled from (d) for the initial and personalised arm templates are compared in Figure 3(f) and (g) respectively, notice using the personalised templates choses a better pose sample.

4 Implementation details

In this section we provide some of the details in our system for producing semi-synthetic CP images and refining the arm templates. We also describe some improvements to the random forest joint detector by Pfister *et al*. [12], which we use to obtain upper body poses. These include a balancing term used during training, addition of a mid lower arm joint and also a fast method for re-training the forest.

Arm template distributions. The colour distributions at each location in an arm template are represented using histograms. We use 10x10x10 bins for the red, green and blue channels respectively. Similarly for torso/head and whole-image templates.

Canonicalising the arm. Arm templates represent upper or lower arms in canonical form. We use a template of size 100 by 21 pixels. Two fixed points in the template represent two different body joints, as shown in Figure 4. The anchor points are used to rotate and anisotropically scale the templates over joint locations for producing the whole-image template. The reverse transforms are used for extracting training windows of arm parts for learning the templates. Note the hand is also included in the lower arm templates in order for hands to be included in the verification function. However, for synthesising CP images, the hand regions are cut off and only the arm part used. This has the effect of leaving the hands of the long sleeved signers intact in the semi-synthetic CP images.

Upper body joint detector. We use a random forest upper body joint detector based on our previous work [12], which detects wrists, elbows, shoulders and head. A failing of this system is that predicted joints can mix-up the left and right wrist. We propose a fix which retains real-time performance of the tracker by including another two joint labels for both left and right mid lower-arms. The mid-arm joints are predicted with higher accuracy than the

wrists (comparison show in Table 1), and can be used for correcting inaccurate wrist joints. Two hypotheses are tested: (i) wrist joints are mixed-up and (ii) wrist joints are correct, verified by scoring each option, where the lowest score wins. Our scoring function is the total least squares error in fitting two lines, one per lower arm, between elbow, mid-arm and wrist locations. Training points for mid-arm locations are computed for the long sleeve videos by taking mid points between provided elbow and wrist locations.

Training the joint detector. Each node in each tree or the random forest consists of a test function. These test functions are learnt recursively from the root node down to the leaf nodes where a distribution over body joint labels is stored. When a training sample S entering the tree is split, data samples satisfying the test S_L go left down the tree and all other samples S_R right. The test function is learnt by maximising the drop in impurity of a split. Pfister *et al*. measured impurity using the Gini entropy, which we augment with a balancing term β. The balancing term helps adjust the ratio of data items going left and right. This assists in reducing tree depth as well as acting as a regulariser to avoid over-fitting. Our drop in impurity measure is formally written as:

$$\triangle i(S) = i(S) - P_L i(S^L) - (1 - P_L) i(S^R) - \beta, \tag{1}$$

where P_L is the fraction of data points that go to the left set and $i(.)$ is the Gini impurity measure. The term $\beta = \lambda \left(\frac{1}{P_L} + \frac{1}{P_R} - |\frac{1}{P_L} - \frac{1}{P_R}| \right)$, where λ controls the weight between the measure of impurity and degree of balancing.

Re-training the forest. It is computationally expensive to train the random forest body joint detector in [12], taking hours of training for only a few hundred images. We found it sufficient to only learn the forest test functions once on the initial semi-synthetic images. When re-training on personalised synthetics, a faster approach can be used as follows: (i) for each tree in the forest set all distributions at the leaf nodes to uniform, (ii) Take a random sample of data items from the personalised semi-synthetic CP images, and (iii) evaluate the leaf node assignment for each data item and re-calculate the distributions at the leaf nodes.

5 Experiments

In this section we test components of our system by judging the detection performance of upper body joints in videos of signers wearing clothes of various sleeve length.

Short sleeve videos. Experiments are conducted on 5 videos containing signers wearing different length sleeves. Sleeve length varies between 1 and 0.2.

Long sleeve videos. For generating semi-synthetic CP images, frames from 10 videos containing a mix of different signers wearing long sleeves are used. Automated ground truth joint locations are provided by Buehler *et al*.'s tracker.

Testing data. Each of the 5 short sleeve videos is split into two sections. The first 60% is used for training and the last 40% is used for testing. The performance of our system is evaluated on 200 diverse frames selected from each testing section, 1000 frames in total. Frames are selected by clustering the CP images of the testing section using k-means into 100 clusters and sampling 2 frames from each cluster. One frame is chosen closest to the cluster centroid, and the second frame chosen within the cluster but furthest from the centroid. This sampling scheme results in a good coverage of the pose space, meaning we do not favour poses which occur more often. Testing frames are manually annotated with joint locations. Accuracy is recorded as an average per joint class over all testing frames.

Evaluation measure. An estimated joint is deemed correctly located if it is within a set distance of d pixels from the ground truth. Accuracy is measured as the percentage of correctly estimated joints.

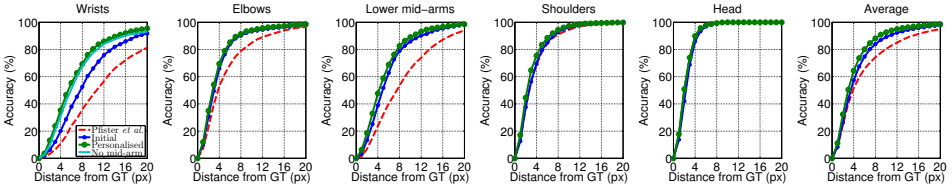


Figure 5: Per-joint average accuracy against allowed pixel distance from ground truth.

Method	Head	Wrists	Elbows	Shoulders	Lower Mid-arms	Average
Pfister *et al.*	94.3	16.4	58.1	75.2	20.8	48.3
Initial without balancing	91.2	23.7	76.8	72.4	42.7	58.1
Initial with balancing	93.5	28.7	76.8	76.8	49.9	62.0
Personalised	**95.8**	**46.6**	**80.7**	**81.3**	**59.6**	**70.2**

Table 1: Average accuracy of per joint estimates. A joint is deemed correct if at most 5 pixels from manual ground truth. Personalising the training shows significant improvement.

Semi-synthetic training data. Five sets of semi-synthetic CP images are generated from the ten long sleeve videos. Each set formed with a sleeve length corresponding in a one-to-one fashion with the short sleeve videos.

Initial forest training. Each tree in the forest is trained on 2,000 CP images sampled from the semi-synthetic training data. Sampling is conducted by clustering the automated ground truth joint locations and sampling from the clusters uniformly. It was found a tree depth of 64 and 8 trees in the forest was optimal.

Personalised semi-synthetic training data. Five sets of arm and torso/head templates are learnt, one set per short sleeve signer video. Each set learnt from 4,000 CP images sampled from the training portion of each video. For the sample/verify step we only sample 10 joint locations per wrist, for all other joints the most confident joints from the forest are chosen.

Personalised forest training. Each tree in the initial forests is retrained using the respective personalised semi-synthetic training data by sampling 2,000 CP images and using them to update distributions at the leaf nodes. We found two iterations of the refinement experiment sufficient, with little improvement thereafter.

Forest balancing. A comparison between using forests with ($\lambda = 0.01$) or without ($\lambda = 0$) a balancing term (see Equation 1) is compared for *initial* forests (trained on initial semi-synthetic data) in Table 1. A modest improvement is seen when using a balancing term in the wrists, lower mid-arm and shoulder joints over not using a balancing term. Therefore, we also choose to use a balancing term in the *personalised* forests.

Lower mid-arm joint. Figure 5 demonstrates using lower mid-arm joint helps to improve the wrist location. It increases accuracy of a wrist joint being within 5 pixels from ground truth from 51% to 55%.

Our method *vs* Pfister *et al*. A comparison in average joint estimation accuracy is shown in Figure 5 between forests trained as in Pfister *et al*. [12] on the long sleeve videos, the *initial* forest (with balancing and mid-arm joints) automatically trained on semi-synthetic images and the updated forest using *personalised* semi-synthetic images. All methods work equally well for head and shoulders with large improvement seen for wrists, elbows and lower mid-arms. Table 1 shows accuracy when a joint is considered correct at only 5 pixels from ground truth (scale bar shown in top left of Figure 1(a) for comparison). Using forests trained on personalised semi-synthetics produces best results for all joint classes. There is a slightly smaller improvement for the initial synthesis approach.

6 Conclusion

We have proposed a method for tracking the upper body pose of signers in TV broadcasts. We achieve this by transferring from existing training material to a new domain, and by automatically personalising the tracker. This is shown to significantly improve pose estimation performance.

In future work we intend to automate the process of recovering side-information about the appearance of a target signer, such as the clothing sleeve length.

Acknowledgements: We are grateful to Patrick Buehler for his generous help. Financial support was provided by the EPSRC grant EP/I012001/1.

References

[1] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *IEEE PAMI*, 2006.

[2] P. Buehler, M. Everingham, and A. Zisserman. Learning sign language by watching TV (using weakly aligned subtitles). In *Proc. CVPR*, 2009.

[3] P. Buehler, M. Everingham, D. P. Huttenlocher, and A. Zisserman. Upper body detection and tracking in extended signing sequences. *IJCV*, 2011.

[4] J. Charles and M. Everingham. Learning shape models for monocular human pose estimation from the Microsoft Xbox Kinect. In *ICCV Workshops*, 2011.

[5] H. Cooper and R. Bowden. Large lexicon detection of sign language. *ICCV Workshops*, 2007.

[6] H. Cooper and R. Bowden. Learning signs from subtitles: A weakly supervised approach to sign language recognition. In *Proc. CVPR*, 2009.

[7] M. Everingham and A. Zisserman. Identifying individuals in video by combining generative and discriminative head models. In *Proc. ICCV*, 2005.

[8] A. Farhadi, D. Forsyth, and R. White. Transfer learning in sign language. In *Proc. CVPR*, 2007.

[9] M. Fergie and A. Galata. Dynamical pose filtering for mixtures of gaussian processes. In *Proc. BMVC*, 2012.

[10] T. Kadir, R. Bowden, E. J. Ong, and A. Zisserman. Minimal training, large lexicon, unconstrained sign language recognition. In *Proc. BMVC*, 2004.

[11] E.J. Ong and R. Bowden. A boosted classifier tree for hand shape detection. In *Face and Gesture*, 2004.

[12] T. Pfister, J. Charles, M. Everingham, and A. Zisserman. Automatic and efficient long term arm and hand tracking for continuous sign language TV broadcasts. In *Proc. BMVC*, 2012.

[13] T. Pfister, J. Charles, and A. Zisserman. Large-scale learning of sign language by watching TV (using co-occurrences). In *Proc. BMVC*, 2013.

[14] L. Pishchulin, A. Jain, M. Andriluka, T. Thormahlen, and B. Schiele. Articulated people detection and pose estimation: Reshaping the future. In *Proc. CVPR*, 2012.

[15] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Proc. ICCV*, 2003.

[16] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proc. CVPR*, 2011.

[17] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE PAMI*, 1998.

[18] M. Sun, P. Kohli, and J. Shotton. Conditional regression forests for human pose estimation. In *Proc. CVPR*, 2012.